

**Updated – version 1.1**

Amendments to study design history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Status | Release Date | Comments |
| 1.1 | Current | February 2023 | Correction to the Master Theorem formula (page 14) |
| 1.0 | Superseded | March 2022 | Original study design. |

Authorised and published by the Victorian Curriculum and Assessment Authority  
Level 7, 2 Lonsdale Street  
Melbourne VIC 3000

ISBN: 978-1-925264-08-1

© Victorian Curriculum and Assessment Authority 2022

No part of this publication may be reproduced except as specified under the *Copyright Act 1968* or by permission from the VCAA. Excepting third-party elements, schools may use this resource in accordance with the [VCAA educational allowance](https://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx). For more information read the [VCAA copyright policy](http://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx).

The VCAA provides the only official, up-to-date versions of VCAA publications. Details of updates can be found on the [VCAA website](http://www.vcaa.vic.edu.au/).

This publication may contain copyright material belonging to a third party. Every effort has been made to contact all copyright owners. If you believe that material in this publication is an infringement of your copyright, please email the [Copyright Officer](mailto:vcaa.copyright@education.vic.gov.au).

Copyright in materials appearing at any sites linked to this document rests with the copyright owner(s) of those materials, subject to the Copyright Act. The VCAA recommends you refer to copyright statements at linked sites before using such materials.

The VCAA logo is a registered trademark of the Victorian Curriculum and Assessment Authority.

|  |
| --- |
| Contact us if you need this information in an accessible format: for example, large print or audio.  Telephone (03) 9032 1635 or email [vcaa.media.publications@education.vic.gov.au](mailto:vcaa.media.publications@education.vic.gov.au) |

Contents

[**Important information 4**](#_Toc35357920)

[**Introduction 5**](#_Toc35357923)

[Scope of study 5](#_Toc35357924)

[Rationale 5](#_Toc35357925)

[Aims 5](#_Toc35357926)

[Structure 6](#_Toc35357927)

[Entry 6](#_Toc35357928)

[Duration 6](#_Toc35357929)

[Changes to the study design 6](#_Toc35357930)

[Monitoring for quality 6](#_Toc35357931)

[Safety and wellbeing 7](#_Toc35357932)

[Employability skills 7](#_Toc35357933)

[Legislative compliance 7](#_Toc35357934)

[Child Safe Standards 7](#_Toc35357934)

[**Assessment and reporting 8**](#_Toc35357935)

[Satisfactory completion 8](#_Toc35357936)

[Levels of achievement 8](#_Toc35357937)

[Authentication 8](#_Toc35357938)

[**Unit 3: Algorithmic problem solving 9**](#_Toc35357939)

[Area of Study 1](#_Toc35357940) [9](#_Toc35357941)

[Area of Study 2 10](#_Toc35357942)

[Area of Study 3 11](#_Toc35357944)

[School-based assessment 12](#_Toc35357946)

[School-assessed Task 13](#_Toc35357947)

[External assessment 13](#_Toc35357948)

[**Unit 4: Principles of algorithmics 14**](#_Toc35357949)

[Area of Study 1 14](#_Toc35357950)

[Area of Study 2](#_Toc35357952) [15](#_Toc35357953)

[Area of Study 3 16](#_Toc35357954)

[School-based assessment 17](#_Toc35357956)

[School-assessed Task 18](#_Toc35357957)

[External assessment 19](#_Toc35357958)

Important information

Accreditation period

Units 3 and 4: 1 January 2023 – 31 December 2026

Implementation of this study commences in January 2023.

Other sources of information

The [*VCAA Bulletin*](https://www.vcaa.vic.edu.au/news-and-events/bulletins-and-updates/bulletin/Pages/index.aspx) is the only official source of changes to regulations and accredited studies. The Bulletin also regularly includes advice on VCE studies. It is the responsibility of each VCE teacher to refer to each issue of the Bulletin. The Bulletin is available as an e-newsletter via [free subscription](https://www.vcaa.vic.edu.au/Footer/Pages/Subscribe.aspx) on the VCAA website.

To assist teachers in developing courses, the VCAA publishes online the Advice for teachers, which includes teaching and learning activities for Units 3 and 4, and advice on assessment tasks and performance level descriptors for School-assessed Coursework in Units 3 and 4.

The current [*VCE and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx) contains essential information on assessment processes and other procedures.

VCE providers

Throughout this study design the term ‘school’ is intended to include both schools and other VCE providers.

Copyright

Schools may use this VCE study design in accordance with the VCAA educational allowance, including making copies for students and use within the school’s secure online teaching and learning environment.   
For details, access the full [VCAA copyright policy](https://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx) on the VCAA website.

Introduction

Scope of study

This study investigates algorithmics, which provides a structured framework for solving real-world, practical problems with computational methods. Algorithmics is fundamental to computer science and software engineering and is essential for understanding the technical underpinnings of the information society. Beyond its use in computing, algorithmics provides a general discipline of rational thought by virtue of the methodical way it approaches problem-solving.

VCE Algorithmics (HESS) examines how information about the world can be systematically represented and how the processes can be made sufficiently explicit and precise so they can be implemented in a computer program. The focus is not on coding but on ‘algorithmic thinking’. Algorithmics covers systematic methods for analysing real-world problems and identifying the salient aspects that need to be modelled as the basis for finding a solution. It explores the design of algorithms to solve these problems, resulting in a powerful approach to manipulating, and reasoning about, structured information.

Mathematical techniques are used to establish crucial properties of algorithms, such as how their performance can be scaled to the size of the problem to be solved. This leads to an understanding of what types of algorithms are able to work efficiently at very large scales. Algorithmics also covers deeper topics in computer science such as the possibility of artificial intelligence, statistical methods of computation, and ethical issues related to both these topics. This investigation of theoretical topics is complemented by the development of skills in a high-level programming language.

Rationale

Computing is central to our society and economy and drives innovation across many fields of human endeavour. Computation has fundamentally transformed the way we conduct science and engineering, as simulation, virtual experiments, and computational analysis and prediction have become indispensable parts of the contemporary scientific method. Computation enables us to make sense of data, whether the data concerns the environment, the economy, health, entertainment, social and organisational structures, or any other sphere of human experience. Algorithmics underpins all computational methods and only through using algorithms can there be full appreciation of their potential and limitations, allowing the development of efficient computational solutions.

VCE Algorithmics (HESS) provides the foundation for studying computer science and software engineering at tertiary level and some universities may offer accelerated pathways to students who have completed the study. The study also provides a conceptual framework for structured and analytical problem-solving in STEM (Science, Technology, Engineering and Mathematics) and other disciplines that benefit from formal reasoning.

Aims

This study enables students to:

* understand the mathematical foundations of computer science
* use symbolic representations and abstraction to formalise real-world information problems
* design algorithms to solve practical information problems, using suitable abstract data types and algorithm design patterns
* investigate the efficiency and correctness of algorithms through formal analysis and empirically through implementation as computer programs
* reason about the mathematical limits of computability
* understand ethical issues relating to data-driven algorithms.

Structure

The study is made up of two units.

* Unit 3: Algorithmic problem-solving
* Unit 4: Principles of algorithmics

Each unit deals with specific content contained in areas of study and is designed to enable students to achieve a set of outcomes for that unit. Each outcome is described in terms of key knowledge and key skills.

Entry

Students must undertake Unit 3 and Unit 4 as a sequence. Units 3 and 4 of a Higher Education scored study (HESS) are designed to the equivalent standard of a first-year university subject. All VCE HESS studies are benchmarked against comparable national and international tertiary curriculums.

The following list identifies important assumed mathematics knowledge that underpins the study design:

* sets and set operations (complement, union, intersection)
* substitution and transposition in linear and non-linear relations
* the construction of tables of values from a given formula
* development of formulas from word descriptions
* sequences and linear relations generated by recursion
* exponents and logarithms
* the ability to produce and interpret numerical plots.

Most of this assumed knowledge is covered in VCE Mathematics Methods Units 1 and 2. Students are expected to be currently enrolled in or have successfully completed VCE Mathematical Methods Units 1 and 2.

Duration

Each unit involves at least 50 hours of scheduled classroom instruction.

Changes to the study design

During its period of accreditation minor changes to the study will be announced in the [*VCAA Bulletin*](https://www.vcaa.vic.edu.au/news-and-events/bulletins-and-updates/bulletin/Pages/index.aspx). The Bulletin is the only source of changes to regulations and accredited studies. It is the responsibility of each VCE teacher to monitor changes or advice about VCE studies published in the Bulletin.

Monitoring for quality

As part of ongoing monitoring and quality assurance, the VCAA will periodically undertake an audit of VCE Algorithmics (HESS) to ensure the study is being taught and assessed as accredited. The details of the audit procedures and requirements are published annually in the [*VCE and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx). Schools will be notified if they are required to submit material to be audited.

Safety and wellbeing

It is the responsibility of the school to ensure that duty of care is exercised in relation to the health and   
safety of all students undertaking the study.

Employability skills

This study offers a number of opportunities for students to develop employability skills. The Advice for teacherscompanion document provides specific examples of how students can develop employability   
skills during learning activities and assessment tasks.

Legislative compliance

When collecting and using information, the provisions of privacy and copyright legislation, such as the Victorian Privacy and Data Protection Act 2014 and Health Records Act 2001, and the federal Privacy   
Act 1988 and Copyright Act 1968, must be met.

Child Safe Standards

Schools and education and training providers are required to comply with the Child Safe Standards made under the Victorian *Child Wellbeing and Safety Act 2005*. Registered schools are required to comply with *Ministerial Order No. 1359 Implementing the Child Safe Standards – Managing the Risk of Child Abuse in Schools and School Boarding Premises*. For further information, consult the websites of the [Victorian Registration and Qualifications Authority](https://www.vrqa.vic.gov.au/childsafe/Pages/Home.aspx), the [Commission for Children and Young People](https://ccyp.vic.gov.au/) and the [Department of Education and Training](https://www2.education.vic.gov.au/pal/child-safe-standards/policy).

Assessment and reporting

Satisfactory completion

The award of satisfactory completion for a unit is based on the teacher’s decision that the student has demonstrated achievement of the set of outcomes specified for the unit. Demonstration of achievement of outcomes and satisfactory completion of a unit are determined by evidence gained through the assessment of a range of learning activities and tasks.

Teachers must develop courses that provide appropriate opportunities for students to demonstrate satisfactory achievement of outcomes.

The decision about satisfactory completion of a unit is distinct from the assessment of levels of achievement. Schools will report a student’s result for each unit to the VCAA as S (satisfactory) or N (not satisfactory).

Levels of achievement

Units 3 and 4

The VCAA specifies the assessment procedures for students undertaking scored assessment in Units 3  
and 4. Designated assessment tasks are provided in the details for each unit in VCE study designs.

The student’s level of achievement in Units 3 and 4 will be determined by School-assessed Coursework (SAC) and one School-assessed Task (SAT) as specified in the VCE study design, and external assessment.

The VCAA will report the student’s level of achievement on each assessment component as a grade from   
A+ to E or UG (ungraded). To receive a study score the student must achieve two or more graded assessments in the study and receive an S for both Units 3 and 4. The study score is reported on a scale of 0–50; it is a measure of how well the student performed in relation to all others who took the study. Teachers should refer to the current [*VCE and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx) for details on graded assessment and calculation of the study score.

Percentage contributions to the study score in VCE Algorithmics (HESS) are as follows:

* Units 3 and 4 School-assessed Coursework: 20 per cent
* Units 3 and 4 School-assessed Task: 20 per cent
* end-of-year examination: 60 per cent.

Details of the assessment program are described in the sections on Units 3 and 4 in this study design.

Authentication

Work related to the outcomes of each unit will be accepted only if the teacher can attest that, to the best   
of their knowledge, all unacknowledged work is the student’s own. Teachers need to refer to the current   
[*VCE and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx) for authentication rules and strategies.

Unit 3: Algorithmic problem-solving

This unit focuses on how algorithms are used for solving complex problems. Algorithms are systematic problem-solving procedures that exist independently of computers. The study of algorithms lies at the heart of computer science and provides the formal foundation for computer programming. Algorithmic problem-solving is a technique that can be applied very broadly in addressing a wide range of complex practical problems.

In Area of Study 1, students develop and apply a range of knowledge and skills to model real-world information problems. In Area of Study 2, students learn how to design algorithms following a variety of simple algorithm design patterns and learn graph algorithms. The programming requirements for Area of Study 2 will be published annually by the VCAA in the [*VCAA Bulletin*](https://www.vcaa.vic.edu.au/news-and-events/bulletins-and-updates/bulletin/Pages/index.aspx). In Area of Study 3, students apply the understanding developed in Areas of Study 1 and 2 to design a solution for a real-world problem that includes both a data representation and algorithm design. Area of Study 3 forms the first part of the School-assessed Task that is completed in Unit 4.

Students are not required to know about the implementation of abstract data types (ADTs), as the main focus of this study is on algorithmic thinking using ADTs rather than on the details of how ADTs are implemented.

Area of Study 1

Data modelling with abstract data types

In this area of study, students develop and apply knowledge and skills in data abstraction. Students consider the structure of information through a study of the definition and properties of abstract data types (ADTs). They select appropriate ADTs and use them to model salient aspects of real-world problems. Students study a variety of collection-based data types, with a particular focus on the graph ADT, which encapsulates a set of nodes along with their interconnections. Students explore how graph ADTs can be applied to network problems, such as social or transport network problems, and planning problems.

Outcome 1

On completion of this unit the student should be able to define and explain the representation of information using abstract data types, and devise formal representations for modelling various kinds of real-world information problems using appropriate abstract data types.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.

Key knowledge

* the motivation for using ADTs
* signature specifications of ADTs using operator names, argument types and result types
* specification and uses of the following ADTs:
* set, list, array, dictionary (associative array)
* stack, queue, priority queue
* graphs, including undirected and directed graphs and unweighted and weighted graphs
* features of graphs, including paths, weighted path lengths, cycles and subgraphs
* categories of graphs, including complete graphs, connected graphs, directed acyclic graphs and trees, and their properties
* modularisation and abstraction of information representation with ADTs
* the structure of decision trees and state graphs

Key skills

* explain the role of ADTs for data modelling
* read and write ADT signature specifications
* use ADTs in accordance with their specifications
* identify and describe properties of graphs
* apply ADTs to model real-world problems by selecting an appropriate ADT and justifying its suitability
* model basic network and planning problems with graphs, including the use of decision trees and state graphs

Area of Study 2

Algorithm design

In this area of study, students learn how to formalise processes as algorithms and to execute them automatically. They use the language of algorithms to describe general approaches to problem-solving and to give precise descriptions of how specific problems can be solved. Students learn how to decompose problems into smaller parts that can be solved independently. This forms the basis of modularisation. Students explore a variety of problem-solving strategies and algorithm design patterns. Students explore example applications of these design patterns and learn about their implications for efficiently solving problems. They learn about recursion as a method for constructing solutions to problems by drawing on solutions to smaller instances of the same problem.

Students are required to implement algorithms as computer programs. The programming language used must explicitly support the ADTs listed in the key knowledge in Area of Study 1 either directly or by using   
a library.

Outcome 2

On completion of this unit the student should be able to define and explain algorithmic design principles, design algorithms to solve information problems using basic algorithm design patterns, and implement the algorithms.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 2.

Key knowledge

* basic structure of algorithms
* pseudocode concepts, including variables and assignment, sequence, iteration, conditionals and functions
* programming language constructs that directly correspond to pseudocode concepts
* conditional expressions using the logical operations of AND, OR, NOT
* recursion and iteration and their uses in algorithm design
* modular design of algorithms and ADTs
* characteristics and suitability of the brute-force search and greedy algorithm design patterns
* graph traversal techniques, including breadth-first search and depth-first search
* specification, correctness and limitations of the following graph algorithms:
* Prim’s algorithm for computing the minimal spanning tree of a graph
* Dijkstra’s algorithm and the Bellman-Ford algorithm for the single-source shortest path problem
* the Floyd-Warshall algorithm for the all-pairs shortest path problem and its application to the transitive closure problem
* the PageRank algorithm for estimating the importance of a node based on its links
* induction and contradiction as methods for demonstrating the correctness of simple iterative and recursive algorithms

Key skills

* interpret pseudocode and execute it manually on given input
* write pseudocode
* identify and describe recursive, iterative, brute-force search and greedy design patterns within algorithms
* design recursive and iterative algorithms
* design algorithms by applying the brute-force search or greedy algorithm design pattern
* write modular algorithms using ADTs and functional abstractions
* select appropriate graph algorithms and justify the choice based on their properties and limitations
* explain the correctness of the specified graph algorithms
* use search methods on decision trees and graphs to solve planning problems
* implement algorithms, including graph algorithms, as computer programs in a very high-level programming language that directly supports a graph ADT
* demonstrate the correctness of simple iterative or recursive algorithms using structured arguments   
  that apply the methods of induction or contradiction

Area of Study 3

Applied algorithms

In this area of study, students combine their knowledge of data modelling and algorithm design to solve real-world problems. Students consider a variety of algorithms and ADTs before selecting a suitable combination. They justify their chosen combination of algorithms and data types relative to other possible choices. Typically the fitness of a chosen combination could be measured in terms of the selection of salient features to achieve an appropriate level of abstraction and the quality of result produced by the algorithm.

Outcome 3

On completion of this unit the student should be able to design suitable solutions for real-world problems that require the integration of algorithms and data types, including the communication of solutions and their justification.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 3.

Key knowledge

* characteristics and applicability of ADTs and algorithm design patterns
* suitability of ADTs and algorithm design patterns for a variety of problem contexts
* combinations of ADTs to meet complex problem requirements
* the application of algorithms to answering real-world problems

Key skills

* describe how complex information can be represented by a combination of ADTs
* select combinations of ADTs and algorithms that are fit for purpose
* justify the suitability of ADTs and algorithm design patterns for particular problems
* communicate the design of data models and algorithms
* explain the interpretation of computed solutions in terms of their meaning to the original real-world problem being solved

School-based assessment

Satisfactory completion

The award of satisfactory completion for a unit is based on whether the student has demonstrated the set of outcomes specified for the unit. Teachers should use a variety of learning activities and assessment tasks to provide a range of opportunities for students to demonstrate the key knowledge and key skills in the outcomes.

The areas of study and key knowledge and key skills listed for the outcomes should be used for course design and the development of learning activities and assessment tasks.

Assessment of levels of achievement

School-assessed Coursework

The student’s level of achievement in Unit 3 will be determined by School-assessed Coursework and a School-assessed Task.

School-assessed Coursework tasks must be a part of the regular teaching and learning program and must not unduly add to the workload associated with that program. They must be completed mainly in class and within a limited timeframe.

Where teachers provide a range of options for the same School-assessed Coursework task, they should ensure that the options are of comparable scope and demand.

The types and range of forms of School-assessed Coursework for the outcomes are prescribed within the study design. The VCAA publishes Advice for teachers for this study, which includes advice on the design   
of assessment tasks and the assessment of student work for a level of achievement.

Teachers will provide to the VCAA a numerical score representing an assessment of the student’s level of achievement. The score must be based on the teacher’s assessment of the performance of each student   
on the tasks set out in the following table.

Contribution to final assessment

School-assessed Coursework for Unit 3 will contribute 12 per cent to the study score.

|  |  |  |
| --- | --- | --- |
| **Outcomes** | **Marks allocated** | **Assessment tasks** |
| **Outcome 1**  Define and explain the representation of information using abstract data types, and devise formal representations for modelling various kinds of real-world information problems using appropriate abstract data types. | **50** | In response to given stimulus material, create one or more designs of a data model using abstract data types to capture the salient aspects of a real-world information problem. |
| **Outcome 2**  Define and explain algorithmic design principles, design algorithms to solve information problems using basic algorithm design patterns, and implement the algorithms. | **50** | In response to given stimulus material:   * create one or more designs of algorithms that apply algorithm design patterns or select appropriate graph algorithms to solve information problems * implement an algorithm. |
| **Total marks** | **100** |  |

School-assessed Task

The student’s level of achievement in Unit 3 Outcome 3, Unit 4 Outcome 1 and Unit 4 Outcome 2 will be assessed through a School-assessed Task. Details of the School-assessed Task for Units 3 and 4 are provided on [page 18](#SchoolassessedTask) of this study design.

External assessment

The level of achievement for Units 3 and 4 is also assessed by an end-of-year examination, which will contribute 60 per cent to the study score.

Unit 4: Principles of algorithmics

This unit focuses on the performance of algorithms and the scope and limitations of algorithms. Students develop the knowledge and skills to identify the resources that an algorithm needs to function efficiently and effectively. In Area of Study 1, students study the efficiency of algorithms and techniques for the formal analysis of algorithms and apply these techniques to an algorithm they designed in Unit 3 Area of Study 3. They also learn about soft limits of computability, namely, problems that can be solved in principle but that cannot be solved for practical problem sizes due to time or space constraints. In Area of Study 2, students learn about a variety of more sophisticated algorithm design patterns and apply their knowledge of these to construct an improved solution for the problem solved in Unit 3 Area of Study 3. In Area of Study 3, students learn about modern data-driven computation and the existence of hard limits of computability, such as problems for which solutions cannot be computed by any computational machinery.

Area of Study 1

Formal algorithm analysis

In this area of study, students investigate the efficiency of algorithms using mathematical techniques. Students learn how some computable problems require such a large amount of resources that in practice it is not possible to solve these exactly for realistic problem sizes. Students examine specific, widely occurring instances of such problems and the reasons why these problems cannot be solved. Students analyse time complexity formally and informally, while they study space complexity as a general concept. Students are not expected to derive the space complexity of algorithms.

Outcome 1

On completion of this unit the student should be able to establish the efficiency of simple algorithms and explain soft limits of computability.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.

Key knowledge

* the concept of classifying algorithms based on their time and space complexity with respect to their input
* techniques for determining the time complexity of iterative algorithms
* the definition of Big-O notation and its application to the worst-case time complexity analysis of algorithms
* recurrence relations as a method of describing the time complexity of recursive algorithms
* the Master Theorem for solving recurrence relations of the form:

where

and its solution:

* examples and common features of algorithms that have time complexities of , , ,   
  , , , and
* the concept of the P, NP, NP-Hard and NP-Complete time complexity classes for problems
* consequences of combinatorial explosions and indicators for them
* the feasibility of NP-Hard problems in real-world contexts

Key skills

* formally analyse the time efficiency of algorithms using Big-*O* notation
* read off a recurrence relation for the running time of a recursive algorithm that can be solved by the Master Theorem or takes the form:, where
* use the stated Master Theorem to solve recurrence relations
* demonstrate how exponentially sized search and solution spaces impose practical limits on computability
* evaluate the suitability of algorithms to particular contexts based on their time or space complexity
* estimate the time complexity of an algorithm by recognising features that are common to algorithms with particular time complexities
* describe characteristics of problems in the P, NP, NP-Hard or NP-Complete time complexity classes, including the consequences for a problem’s feasibility of it belonging to one of these classes

Area of Study 2

Advanced algorithm design

In this area of study, students examine more advanced algorithm design patterns. Students learn how to select algorithmic approaches from a wider range of options, depending on the structure of the problem that is being addressed. They investigate how some problems are solvable in principle while being intractable in practice. They explore examples of such problems with real-world relevance and learn how such problems can be tackled by computing near-optimal solutions.

Outcome 2

On completion of this unit the student should be able to solve a variety of information problems using algorithm design patterns and explain how heuristics can address the intractability of problems.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 2.

Key knowledge

* the binary search algorithm
* divide and conquer algorithms that have linear time divide and merge steps, including mergesort and quicksort
* dynamic programming algorithms that require no more than a single dimension array for storage, including the Fibonacci numbers and change-making problem
* tree search by backtracking and its applications
* the application of heuristics and randomised search to overcoming the soft limits of computation, including the limitations of these methods
* hill climbing on heuristic functions, the A\* algorithm and the simulated annealing algorithm
* the graph colouring, 0-1 knapsack and travelling salesman problems and heuristic methods for solving them

Key skills

* apply the divide and conquer, dynamic programming and backtracking design patterns to design algorithms and recognise their usage within given algorithms
* develop different algorithms for solving the same problem, using different algorithm design patterns, and compare their suitability for a particular application
* apply heuristics methods to design algorithms to solve computationally hard problems
* explain the application of heuristics and randomised search approaches to intractable problems, including the graph colouring, 0-1 knapsack and travelling salesman problems

Area of Study 3

Computer science: past and present

In this area of study, students examine the emergence of computer science as a field and the philosophical and technical ideas that support the emergence of modern artificial intelligence (AI). They explore how the quest to develop methods for mathematical proof led to the proof that there exist problems that may not be computed automatically. Students investigate how machine learning algorithms learn from data and engage with several conceptions of artificial intelligence and whether it is possible. They examine and discuss some of the ethical issues posed by the application of data-driven algorithms. Students are not required to produce proofs or formal explanations concerning undecidability.

Outcome 3

On completion of this unit the student should be able to explain the historical context for the emergence of computer science as a field and discuss modern machine learning techniques and the philosophical issues they raise.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 3.

Key knowledge

* the historical connections between the foundational crisis of mathematics in the early 20th century and the origin of computer science, including Hilbert and Ackermann’s Entscheidungsproblem and its resolution by Church and Turing
* characteristics of a Turing machine
* the concept of decidability and the Halting Problem as an example of an undecidable problem
* implications of undecidability for the limits of computation
* philosophical conceptions of artificial intelligence, including the Turing Test, weak AI and strong AI
* Searle’s Chinese Room Argument, including standard responses both for and against
* the concept of training algorithms using data
* the concepts of model overfitting and underfitting
* support vector machines (SVM) as margin-maximising linear classifiers, including:
* the geometric interpretation of applying SVM binary classification to one- or two-dimensional data
* the creation of a second feature from one-dimensional data to allow linear classification
* neural networks, including:
* the structure of multi-layer perceptron neural networks
* the evaluation of outputs using forward propagation
* training neural networks by using iterative improvement of the edge weights to reduce the output error
* the factors leading to a resurgence in neural networks in the late 20th century
* ethical issues related to artificial intelligence and data-driven algorithms, including transparency, accountability, bias and machine ethics

Key skills

* explain the historical context for the emergence of computer science as a field
* describe the general structure of a Turing machine
* demonstrate the existence of hard limits of computability using the Halting Problem
* describe and compare the Turing Test, strong AI and weak AI as conceptions of artificial intelligence
* describe the Chinese Room Argument and mount an argument for or against it
* explain, at a high level, how data-driven algorithms can learn from data
* explain the optimisation objectives for training SVM and neural network binary classifiers
* explain how higher dimensional data can be created to allow for linear classification
* describe the structure of a multi-layer perceptron neural network
* evaluate the output of a small multi-layer perceptron neural network using forward propagation
* explain the consequences of model overfitting or underfitting
* explain and discuss ethical issues related to artificial intelligence and data-driven algorithms

School-based assessment

Satisfactory completion

The award of satisfactory completion for a unit is based on whether the student has demonstrated the set of outcomes specified for the unit. Teachers should use a variety of learning activities and assessment tasks to provide a range of opportunities for students to demonstrate the key knowledge and key skills in the outcomes.

The areas of study and key knowledge and key skills listed for the outcomes should be used for course design and the development of learning activities and assessment tasks.

Assessment of levels of achievement

School-assessed Coursework

The student’s level of achievement in Unit 4 will be determined by School-assessed Coursework and a School-assessed Task.

School-assessed Coursework tasks must be a part of the regular teaching and learning program and must not unduly add to the workload associated with that program. They must be completed mainly in class and within a limited timeframe.

Where teachers provide a range of options for the same School-assessed Coursework task, they should ensure that the options are of comparable scope and demand.

The types and range of forms of School-assessed Coursework for the outcomes are prescribed within the study design. The VCAA publishes Advice for teachers for this study, which includes advice on the design of assessment tasks and the assessment of student work for a level of achievement.

Teachers will provide to the VCAA a numerical score representing an assessment of the student’s level of achievement. The score must be based on the teacher’s assessment of the performance of each student on the tasks set out in the following table.

Contribution to final assessment

School-assessed Coursework for Unit 4 will contribute 8 per cent to the study score.

|  |  |  |
| --- | --- | --- |
| **Outcomes** | **Marks allocated** | **Assessment tasks** |
| **Outcome 3**  Explain the historical context for the emergence of computer science as a field and discuss modern machine learning techniques and the philosophical issues they raise. | **50** | Select at least one task from the following:   * a response to a case study or stimulus material * a written report * an annotated visual report * an oral report * structured questions. |
| **Total marks** | **50** |  |

School-assessed Task

The student’s level of achievement in Unit 3 Outcome 3, Unit 4 Outcome 1 and Unit 4 Outcome 2 will be assessed through a School-assessed Task.

Contribution to final assessment

The School-assessed Task contributes 20 per cent to the study score.

|  |  |
| --- | --- |
| **Outcomes** | **Assessment tasks** |
| **Unit 3**  **Outcome 3**  Design suitable solutions for real-world problems that require the integration of algorithms and data types, including the communication of solutions and their justification. | The design of a data model and algorithm combination to solve a real-world/applied problem, including:   * a specification of the problem * a consideration of multiple solution options * the selection of a suitable, coherent, clear and fit-for-purpose solution   **AND**  A formal time complexity analysis of the designed algorithm for the applied problem and an explanation of the consequences of these results on the algorithm’s real-world application.  **AND**  A design of an improved data model and algorithm combination to solve the applied problem, including:   * the selection of an efficient, coherent and fit-for-purpose solution * a time complexity analysis * a comparison to the original solution. |
| **Unit 4**  **Outcome 1**  Establish the efficiency of simple algorithms and explain soft limits of computability. |
| **Unit 4**  **Outcome 2**  Solve a variety of information problems using algorithm design patterns and explain how heuristics can address the intractability of problems. |

External assessment

The level of achievement for Units 3 and 4 is also assessed by an end-of-year examination.

Contribution to final assessment

The examination will contribute 60 per cent to the study score.

End-of-year examination

Description

The examination will be set by a panel appointed by the VCAA. All the key knowledge and key skills that underpin the outcomes in Units 3 and 4 are examinable.

Conditions

The examination will be completed under the following conditions:

* Duration: 2 hours
* Date: end-of-year, on a date to be published annually by the VCAA
* VCAA examination rules will apply. Details of these rules are published annually in the [*VCE and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx)
* The examination will be marked by assessors appointed by the VCAA.

Further advice

The VCAA publishes specifications for all VCE examinations on the VCAA website. Examination specifications include details about the sections of the examination, their weighting, the question format(s) and any other essential information. The specifications are published in the first year of implementation of the revised Unit 3 and 4 sequence together with any sample material.